

### **REMARKS**

The Office Action mailed May 1, 2007 considered claims 9-21 and 29-46. Claims 29-32 were rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter. Claims 9-16, 18-21 and 29-46 were rejected under 35 U.S.C. 102(e) as being anticipated by Peng (US 6,928,467) hereinafter *Peng*. Claim 17 was rejected under 35 U.S.C. 103(a) as being unpatentable over *Peng* in view of LaRue et al. (US 2002/0133508) hereinafter *LaRue*.<sup>1</sup>

By this paper, claims 9, 29, 33, and 46 have been amended, claim 42 has been cancelled, such that claims 9-21, 29-40, and 42-46 remain pending in the application, of which, only claims 9, 29, 32 and 33 are the only independent claims in the application.

As a preliminary Applicants would like to thank the Examiner for the courtesies extended during the interview of June 19, 2007. The substance of that interview is included herein.

#### **Rejections Under 35 USC § 101**

Claims 29-32 were rejected under 35 USC § 101 as being directed to non-statutory subject matter. In particular, the Office Action states that "[i]ndependent claim 29 recites the limitation 'wherein the convey changes message is used to determine whether or not a change represented in the change argument should be applied to the second replica.'" The Office Action asserts that "[t]he claim fails to produce a tangible results when it is determined that the convey changes message indicates that a change should not be applied to the second replica."

Claim 29 has been amended to recite "wherein the made-with-knowledge argument is used to determine to selectively apply a change represented in the change argument to the second replica." Applicants believe that this limitation positively recites a tangible result.

#### **Rejections Under 35 USC § 102**

The invention is generally directed to implementing a synchronization method for replicas in a sync community. Various embodiments are directed to conveying messages indicating changes of which a replica is aware. This can be useful to accomplish several tasks such as reducing the number of changes that are exchanged as changes which have already been applied do not need to be resent. Also, a replica can advertise changes that it has available such that other replicas can then request needed changes from the advertising replica. Further various

---

<sup>1</sup> Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

conflict resolution tasks can be performed by knowing a version and replica for a particular change.

Illustratively, each of the independent claims recites sending or including knowledge "including information representing a plurality of changes that the first replica is aware of by including information representing a change ID for each change that the first replica is aware of, wherein *each change ID includes a replica ID associated with the change and a version specific to a specific change*, wherein *knowledge of at least two or more changes is included in a vector, the vector representing a plurality of change IDs*" or similar limitations. The claims further clarify that the vector claimed includes one or more replica ID elements, and one or more magnitudes, where the magnitudes represent the number of changes in the vector.

The use of a vector which includes a representation of two or more change IDs is not shown by *Peng*. Rather, *Peng* shows that version information can be implemented in one of three way including a dirty bit, an update sequence number, and a version vector. Notably as will be discussed below, the version vector described by *Peng* differs significantly from the vector defined in the claims of the present application.

The version vector described by *Peng* shows that version information is represented as a replica ID and a time stamp. See *Peng* at col. 5, lines 14-25. In other words, the vector represented is identified by the replica ID and has a magnitude defined in terms of time, and not in terms of changes. Only a single change can be deduced from the version vector disclosed by *Peng*. The time stamp indicates the last known time that an object store has updated at least one of its objects. *Id.* Thus, it would appear that version vector described by *Peng* at the very most only shows an indicator of the last time a single change was made. This is supported by *Peng* which states that the version vector corresponds to an "object store replica [that is] known to have performed one update to at least one object associated with that object store replica." *Peng* at 20-24. Thus, the vector only includes information that one update was performed, but does not include any specific identifying information about updates performed at a given replica previous to the one identified update. This does not convey the amount of knowledge claimed in the present application, where a vector includes multiple change IDs indicating knowledge about multiple changes individually.

*Peng* also discloses a dirty bit. See *Peng* at col. 4, line 64. However, the dirty bit disclosed by *Peng* is not in vector form as claimed in the present application. Rather, the dirty

bits are included on an object by object basis. See e.g. col. 11, lines 63-68. A dirty bit simply "indicates whether an associated object is in a different state than the object's replica at another object store" but does not include identifiers for at least two or more changes in vector form. See *Peng* at col. 4, line 65-67.

*Peng* further discloses an update sequence number. *Peng* at col. 4, lines 64-65. However, the update sequence number is "an integer that increments with time" and is used to "compare an object replica to the object and to other replicas to determine the age of the object replica relative to the object and to the other replicas." Thus, the update sequence number is not a vector, and even assuming *arguendo* that it is a vector, it does not represent a plurality of change IDs as is recited by the claims of the present application.

*LaRue* does not compensate for the deficiencies of *Peng*. Rather, *LaRue* illustrates a method of preventing circular synchronization. To accomplish this, *LaRue* sends information mapping fields to be synchronized to fields at a third party. See *LaRue* at [0055]. However, *LaRue* fails to teach or suggest what is now recited by the claims of the present application.

While the dependent claims do not need to be discussed due to the patentability of the independent claims over the art cited, and the patentability of the dependent claims by virtue of their dependence on the independent claims, Applicants would like to point out a number of the dependent claims and request detailed reconsideration.

For example, claim 30 recites "storing the convey changes message on a removable medium and transporting the removable medium to the second replica...." To show this element cites to column 1, lines 31-47 of *Peng* which simply shows that data can be stored on various media, but does not show a convey changes message on any media. Thus 35 USC 102 is inappropriate for this claim. Claim 30 highlights that where knowledge is included in the convey change message, synching can be accomplished beyond traditional device to device connected synching. Rather, synching can actually be accomplished by conveying changes by transporting them on a removable medium. *Peng* does not show this disconnected method of transmitting changes.

Claim 42 recites that the knowledge includes an exception list. To show this element, the Office Action cites to *Peng* at col. 17, lines 15-20 and col. 18, lines 51-58. However, this portion of *Peng* only shows various encoding methods. No exception list is shown. To help in facilitating understanding the exception list concept, attention is directed to Figure 7C of the

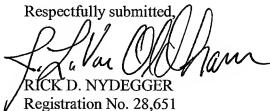
present application which shows an example exception list at 716. The knowledge vector 714 includes change IDs that can be represented continuously. The exception list 714 illustrates how when there is a break in the continuum, knowledge of changes and change IDs are still able to be represented. This illustrates just one embodiment including "when knowledge of changes cannot be continuously represented by the vector, the exception list including additional change IDs for changes outside the range of the vector" as recited by claim 42. However, the office action does not directly point out any teaching showing what is recited by claim 42.

In view of the foregoing, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice. Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at 801-533-9800.

Dated this 2<sup>nd</sup> day of July, 2007.

Respectfully submitted,



RICK D. NYDEGGER  
Registration No. 28,651  
J. LAVAR OLDHAM  
Registration No. 53,409  
Attorneys for Applicant  
Customer No. 47973